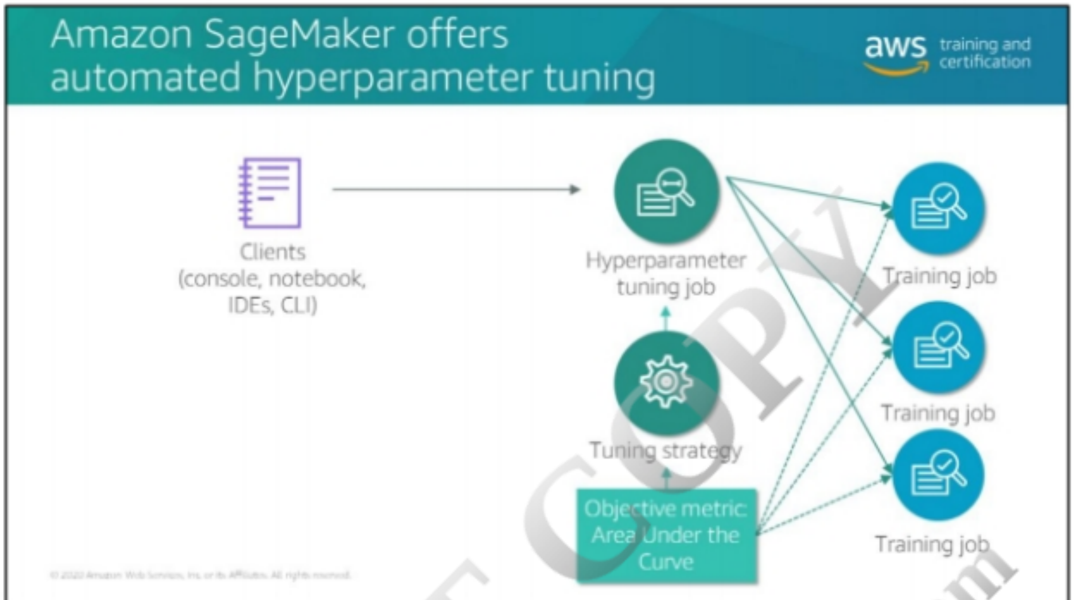
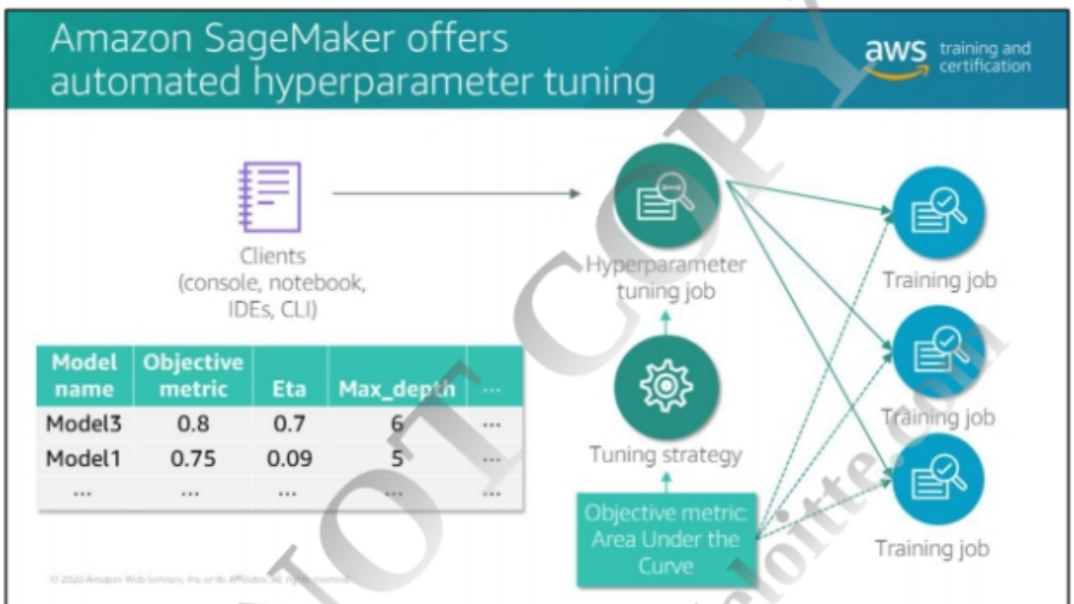


Then there's *automated hyperparameter tuning*, which uses methods like gradient descent, Bayesian optimization, and evolutionary algorithms to conduct a guided search for the best hyperparameter settings.

Amazon SageMaker lets you perform automated hyperparameter tuning. Amazon SageMaker automatic model tuning, also known as *hyperparameter tuning*, finds the best version of a model by running many training jobs on your dataset using the algorithm and ranges of hyperparameters that you specify. It then chooses the hyperparameter values that result in a model that performs the best, as measured by a metric that you choose. It uses Gaussian Process regression to predict which hyperparameter values might be most effective at improving fit. It also uses Bayesian optimization to balance exploring the hyperparameter space and exploiting specific hyperparameter values when appropriate. And importantly, automatic model tuning can be used with the Amazon SageMaker built-in algorithms, pre-built deep learning frameworks, and bring-your-own-algorithm containers.



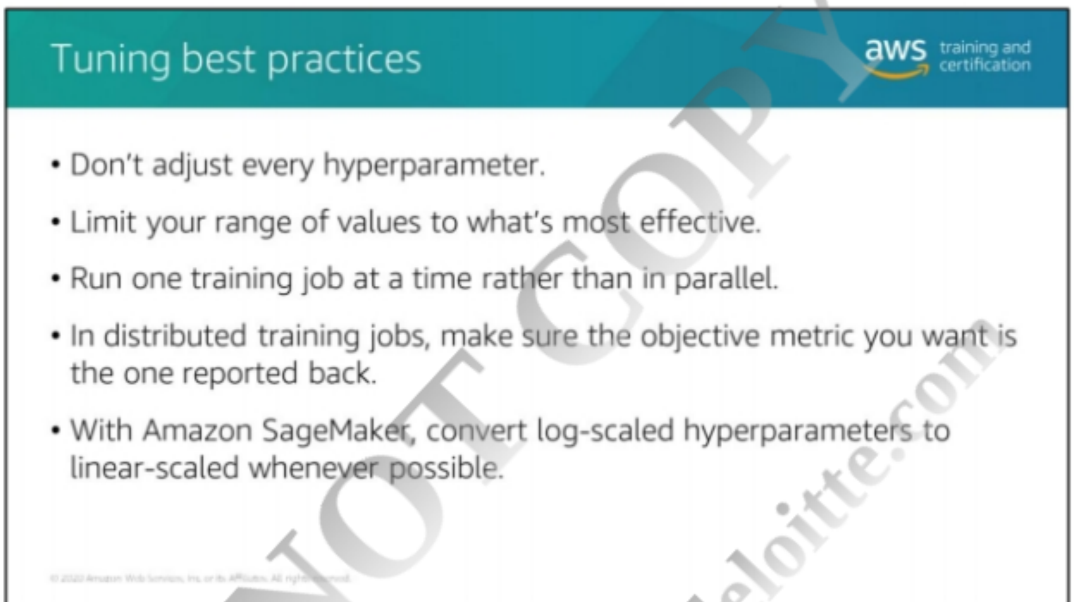
For example, suppose that you want to solve a binary classification problem on a fraud dataset. Your goal is to maximize the area under the curve (AUC) metric of the algorithm by training a Linear learner Algorithm model. You don't know which values of the learning_rate, beta_1, beta_2, and epochs to use to train the best model. To find the best values for these hyperparameters, you can specify ranges of values that Amazon SageMaker hyperparameter tuning searches to find the combination of values that results in the training job that performs the best as measured by the objective metric that you chose. Hyperparameter tuning launches training jobs that use hyperparameter values in the ranges that you specified, and returns the training job with the highest AUC.

Hyperparameter tuning

aws training and certification

Note: Tuning doesn't always improve your model.

Hyperparameter tuning might not necessarily improve your model. It is an advanced tool for building machine solutions, and, as such, should be considered part of the scientific method process. When you build complex machine learning systems like deep learning neural networks, exploring all of the possible combinations is impractical



To improve optimization, use the following guidelines when you create hyperparameters.

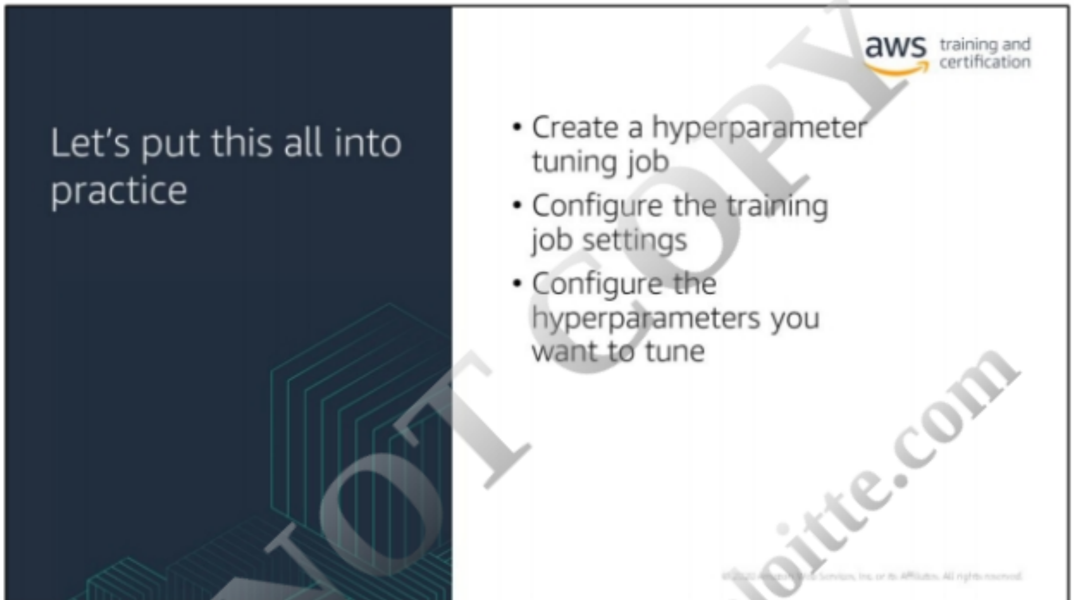
- Limit the number of hyperparameters to the ones that you think would most likely give you good results rather than using all hyperparameters.

- The range of values for hyperparameters that you choose to search can significantly affect the success of hyperparameter optimization. Although you might want to specify a very large range that covers every possible value for a hyperparameter, you will get better results by limiting your search to a small range of values. If you get the best metric values within a part of a range, consider limiting the range to that part.

- During hyperparameter tuning, Amazon SageMaker attempts to figure out if your hyperparameters are log-scaled or linear-scaled. Initially, it assumes that hyperparameters are linear-scaled. If they should be log-scaled, it might take some time for Amazon SageMaker to discover that. If you know that a hyperparameter should be log-scaled and can convert it yourself, doing so could improve hyperparameter optimization.

- Running more hyperparameter tuning jobs concurrently gets more work done quickly, but a tuning job improves only through successive rounds of experiments. Typically, running one training job at a time achieves the best results with the least amount of compute time.

- When a training job runs on multiple instances, hyperparameter tuning uses the last-reported objective metric from all instances of that training job as the value of the objective metric for that training job. Design distributed training jobs so that you get they report the objective metric that you want.



To get started with Amazon SageMaker Hyperparameter Optimization let's get into the service. We'll create our tuning job, configure our training job settings, and then configure the hyperparameters we want to optimize.


aws training and certification
Demo 4: Amazon SageMaker hyperparameter optimization

Summary

Summary

- Explain feature engineering
- List some of the techniques to handle numerical features.
- List some scaling transformation techniques
- Explain the three categories of hyperparameters
- Explain tuning best practices

Feature engineering:

- Feature engineering is the science (and art) of extracting more information from existing data in order to improve your model's prediction power and help your model learn faster. You are not adding any new data during featuring engineering, but, rather, you are making the data you already have more useful. This process often relies on domain knowledge of the data to engineer more effective features.

Some techniques to handle numerical features:

- Logarithmic transformation, square or cube, binning and scaling

Some scaling transformation techniques:

- Mean/variance standardization
- MinMax scaling
- Maxabs scaling
- Robust scaling
- Normalizer

The three categories of hyperparameters:

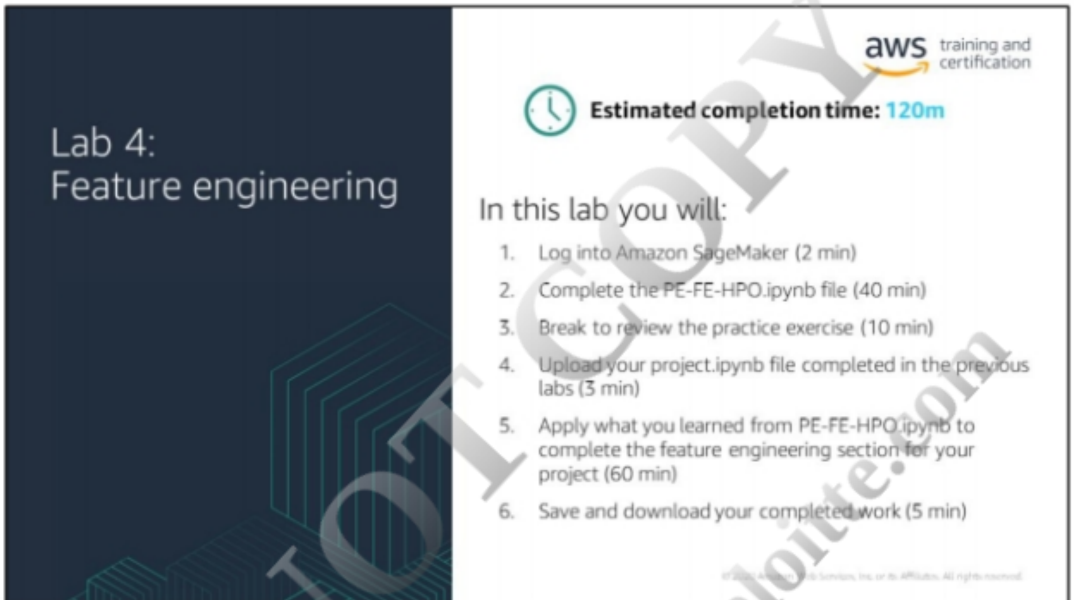
- Model hyperparameters – help define the model itself
- Optimizer hyperparameters – define how the model learns the patterns based
- Data hyperparameters – define data augmentation techniques like cropping or resizing for image-related problems

Tuning best practices:

- To improve optimization, use the following guidelines when you create hyperparameters.
- Limit the number of hyperparameters to the ones that you think would most likely give you good results rather than using all hyperparameters.
- The range of values for hyperparameters that you choose to search can significantly affect the success of hyperparameter optimization. Although you might want to specify a very large range that covers every possible value for a hyperparameter, you will get better results by limiting your search to a small range of values. If you get the best metric values within a part of a range, consider limiting the range to that part.
- During hyperparameter tuning, Amazon SageMaker attempts to figure out if your hyperparameters are log-scaled or linear-scaled. Initially, it assumes that hyperparameters are linear-scaled. If they should be log-scaled, it might take some time for Amazon SageMaker to discover that. If you know that a hyperparameter should be log-scaled and can convert it yourself, doing so could improve hyperparameter optimization.
- Running more hyperparameter tuning jobs concurrently gets more work done quickly, but a tuning job improves only through successive rounds of experiments. Typically, running one training job at a time achieves the best results with the least amount of compute time.
- When a training job runs on multiple instances, hyperparameter tuning uses the last-reported objective metric from all instances of that training job as the value of the objective metric for that training job. Design distributed training jobs so that you get they report the objective metric that you want.

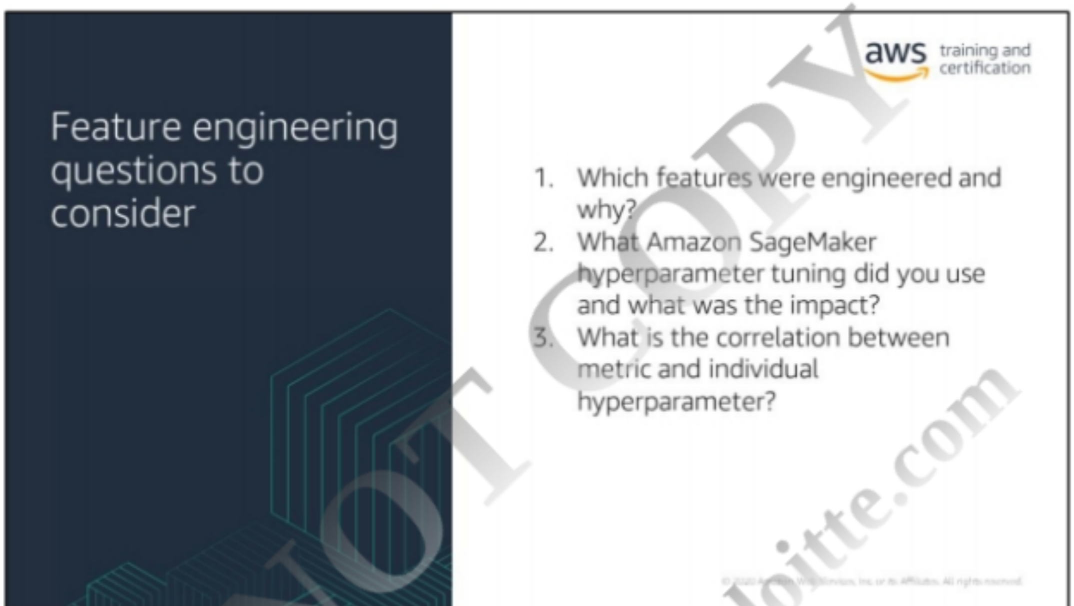
aws training and certification

Lab 4: Feature Engineering

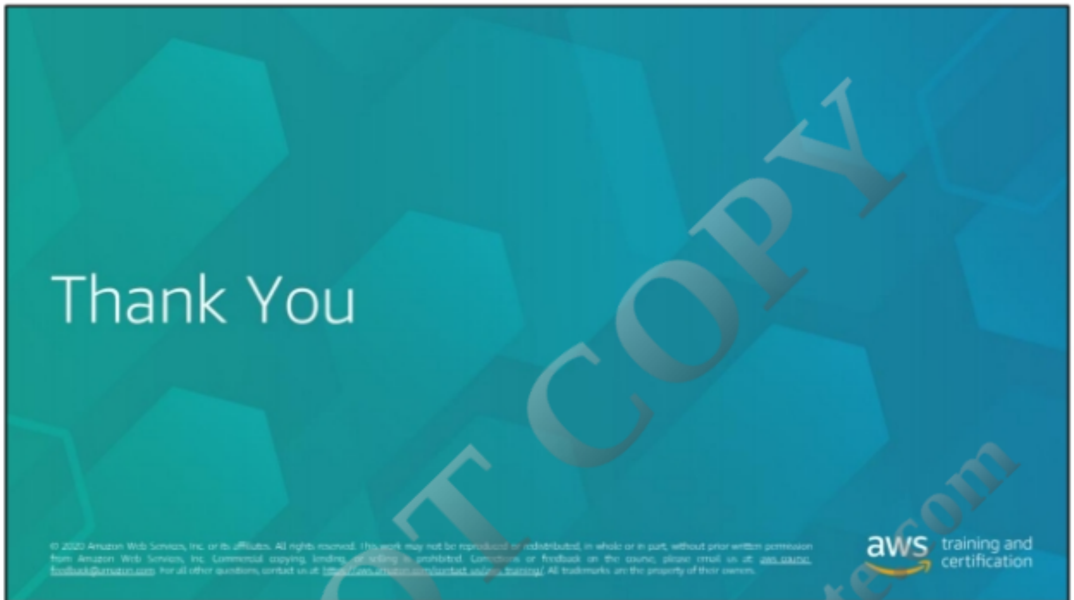


You'll now have an opportunity to gain hands-on experience with Amazon SageMaker hyperparameter optimization by tuning your own training job.

Note: Be sure to take the break to review the practice exercise.



As you work through feature engineering for your model, be sure to track any relevant findings in your project template.


Thank You
aws training and certification